Reconfigurable Computing Architecture for Linux

Vince Bridgers & Yves Vandervennet

October 13th, 2016

Agenda

- Brief Introduction to Heterogeneous Computing
- Broad range of Systems Structures
- Some interesting use cases
- Heterogeneous Computing Architecture for Linux

The Programmer's Challenge …

"The way the processor industry is going, is to add more and more cores, but nobody knows how to program those things. I mean, two yeah; four not really; eight, forget it." - Steve Jobs

Objectives - Define Reconfigurable Compute Architecture for Linux

- Use Open Source to define and develop a reference implementation/platform encouraging collaboration and innovation
  - Application developers and user will have a consistent experience using these tools and using Linux as the Operating System platform
  - Example: in-kernel FPGA manager framework
- Accelerate adoption of offload technologies in embedded, datacenter, cloud, and embedded systems, providing good developer and user experiences
- Define the interfaces such vendor specific innovations can be implemented in User Mode applications – the kernel bits can be thought of as plumbing.
- Support as many offload technologies and system types as possible.

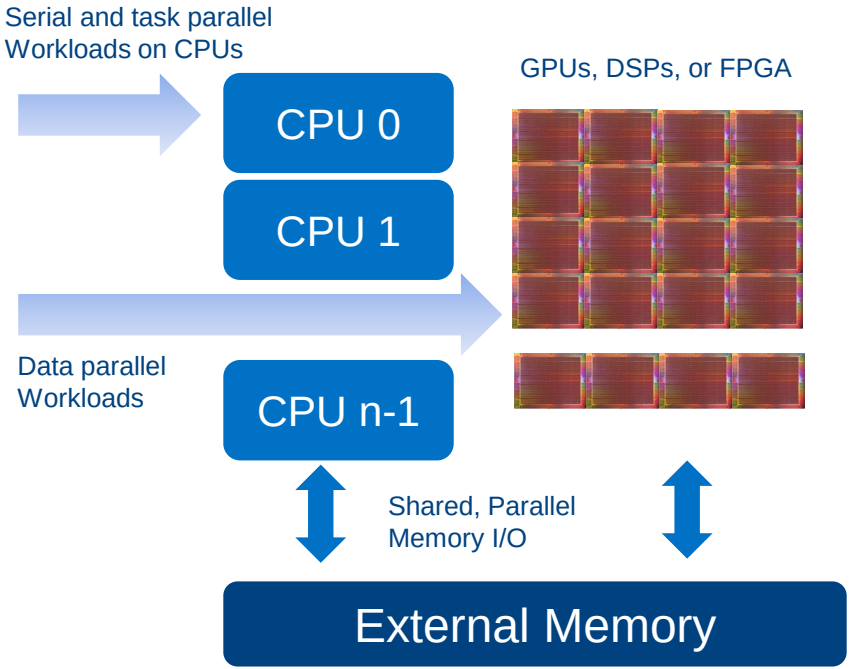
Heterogeneous System Architecture Review

Reconfigurable Computing

- **Takes advantage of CPUs for serial and task parallel workloads.**
- **CPUs can be any architecture (x86, ARM, etc)**
- **Takes advantages of computing elements that are good at data parallel workloads.**

Computing elements

- **Can be GPUs, DSPs, or FPGA**
- **Interconnect to computing elements can be PCIe, AXI, etc**
- **The “reconfigurable” part comes in since elements can be re-provisioned to solve particular problems based on software, firmware, or synthesized logic.**



Heterogeneous System Architecture Review

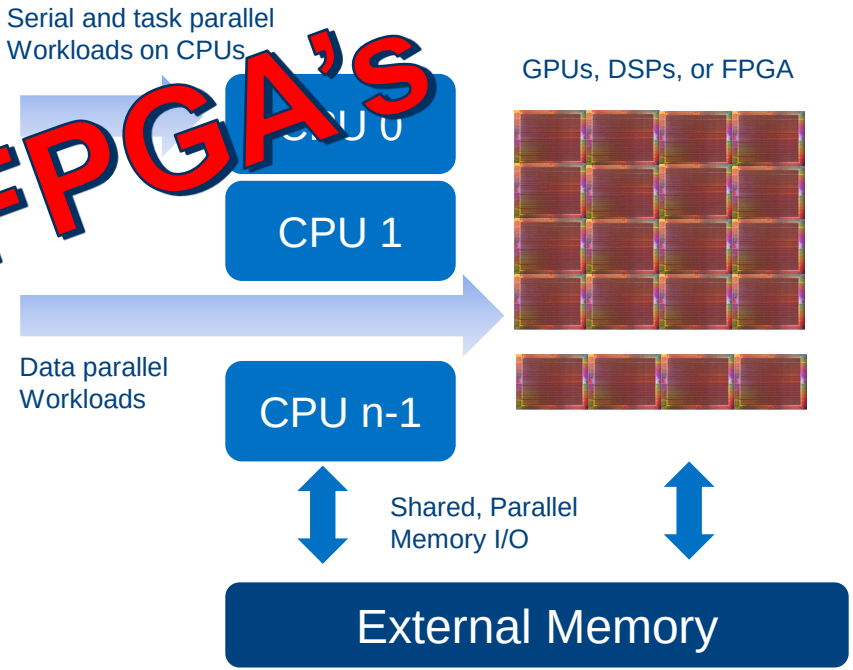
Reconfigurable Computing

- **Takes advantage of CPUs for serial and task parallel workloads.**
- **CPUs can be any architecture (x86, ARM, etc)**
- **Takes advantages of computing elements that are good at data parallel workloads.**

Computing elements

- **Can be GPUs, DSPs, or FPGA**
- **Interconnect to computing elements can be PCIe, AXI, etc**
- **The "reconfigurable" part comes in since elements can be re-provisioned to solve particular problems based on software, firmware, or synthesized logic.**

Focus: FPGA's



Computing elements: FPGA's

FPGA = Field Programmable Gate Array

- Array of programmable logic blocks, aka Fabric
  - Generic elements providing latches/flip-flops and gates
  - Specialized elements like multipliers, transceivers
- Designed to be configured after manufacturing
  - HDL's are used to describe the HW functions
  - HDL is compiled to a bit stream
- A Bit Stream is used to program the FPGA
  - Typically, configured at power-on
  - Means of Configuration examples: from flash, over PCIe

Computing elements: FPGA's (cont'd)

FPGA = Field Programmable Gate Array

- Full reconfiguration
  - The entire FPGA is reprogrammed, functions and I/O
- Partial reconfiguration
  - A limited area of the FPGA is reprogrammed

Computing elements: FPGA's (cont'd)

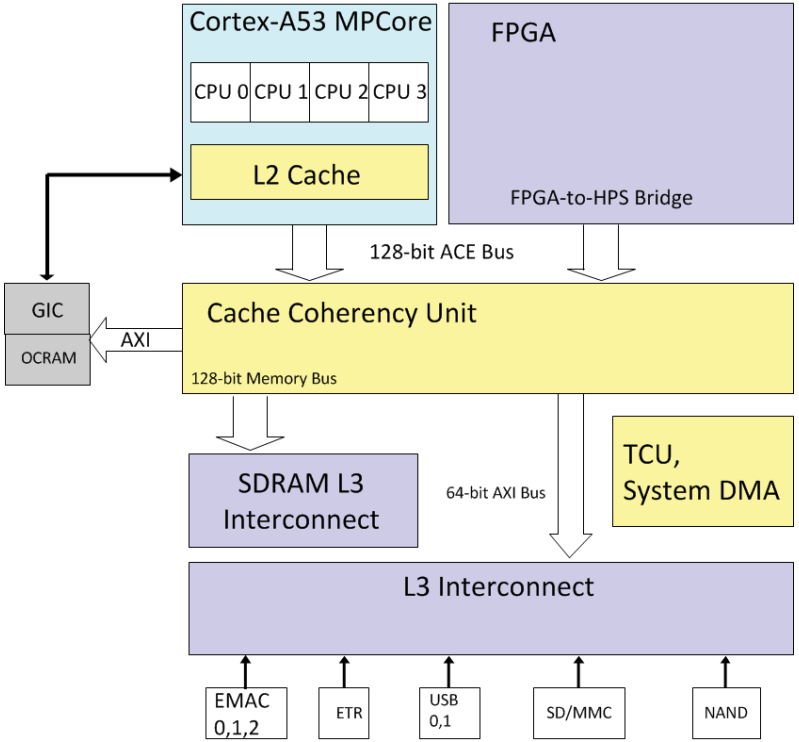
Typical Workflow:

Example of use cases:

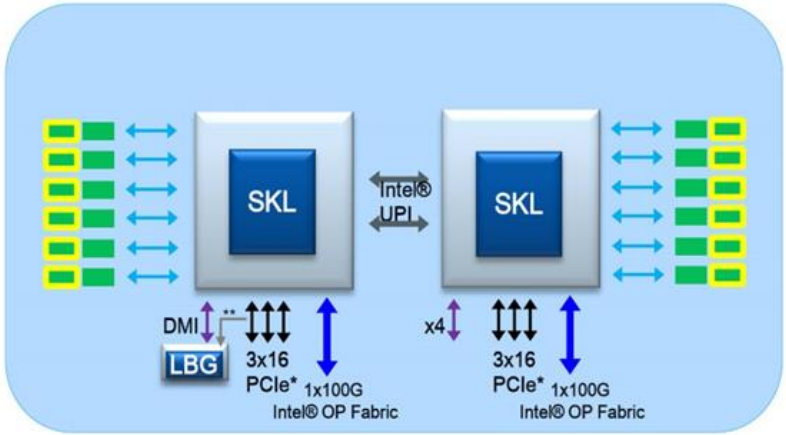
- Industrial: motor control, Industrial Ethernet
- Multimedia/Broadcast: video/image processing
- Telecommunication: Ethernet switches, packet process offload
- HPC: search engine acceleration, complex acceleration algorithms

Typical System Structures – Embedded Systems, Client/Server Systems

Embedded System



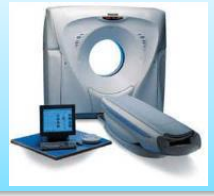
Small Server System



Reconfigurable Computing Use Cases

Multimedia

- HD video processing
- VOD
- Image recognition (CNN)

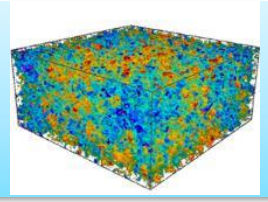


Medical

- MRI
- CT
- PET

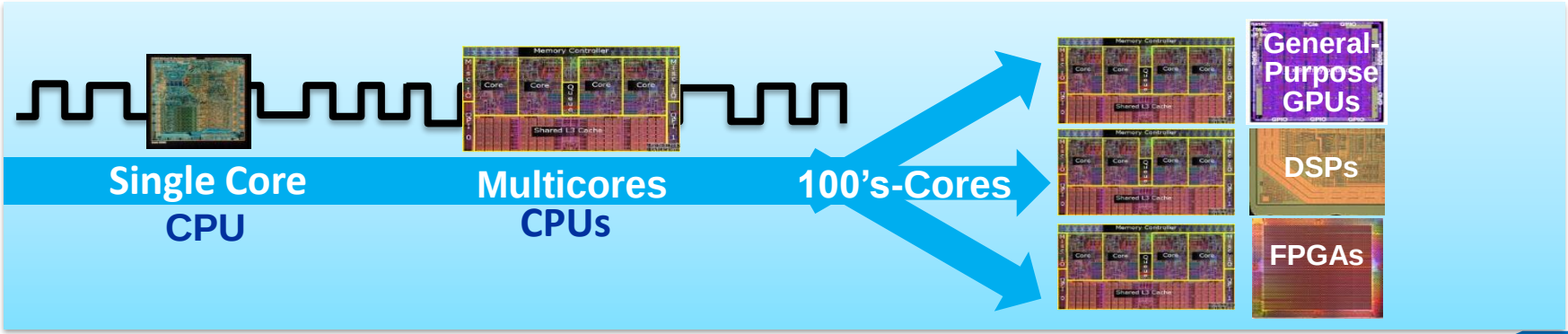
Radar

- Pulse Doppler radar
- STAP
- Passive radar



High-Performance Computing

- Machine Learning
- Climate modeling
- Financial modeling



Existing Technologies supporting Reconfigurable Computing

An important component to support reconfigurable computing are the software tools and development flow supporting Reconfigurable Computing.

Linux Kernel FPGA Manager

- A starting point for FPGA programming and reconfiguration in embedded systems

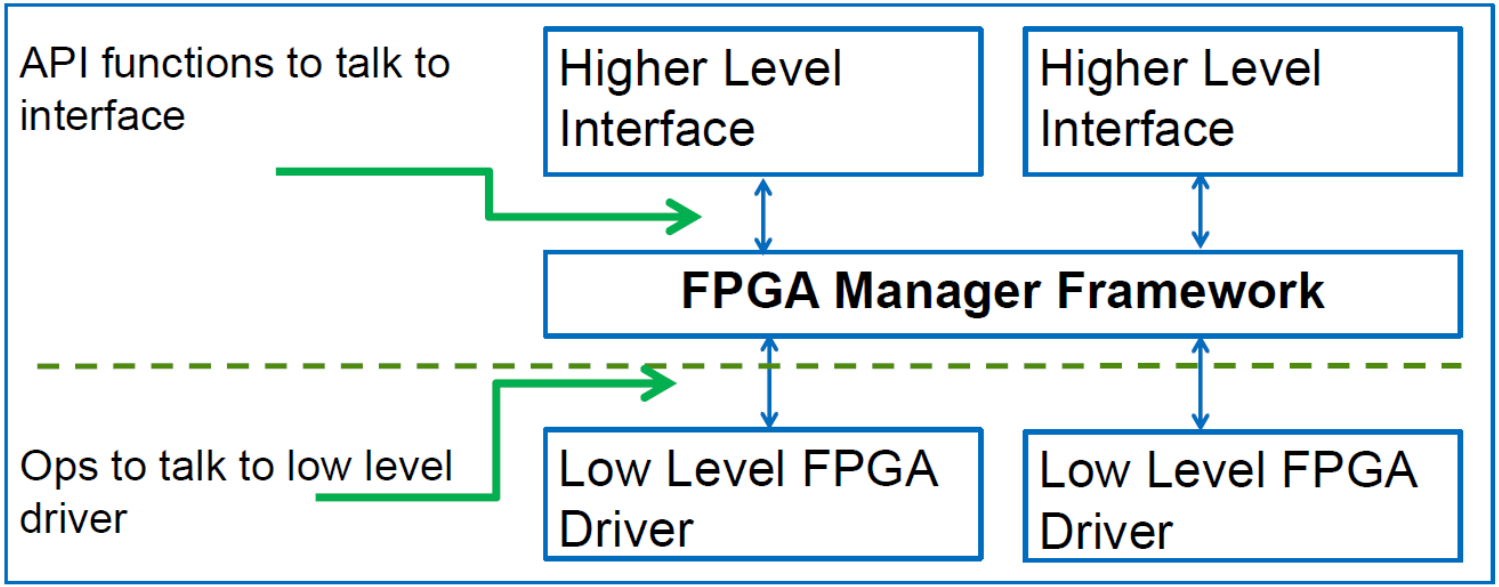
OpenCL

- OpenCL is a tool to develop complete software applications that leverage offload elements as accelerators
- OpenCL targets GPUs, CPUs, and FPGAs to partition task parallel and data parallel workloads across available resources.
- Most implementations today are vendor specific and are “vertical” implementations with no standardization of OS plumbing.

High Level Synthesis

- An important piece of the complete puzzle for reconfigurable computing, but not that important for today’s discussion.

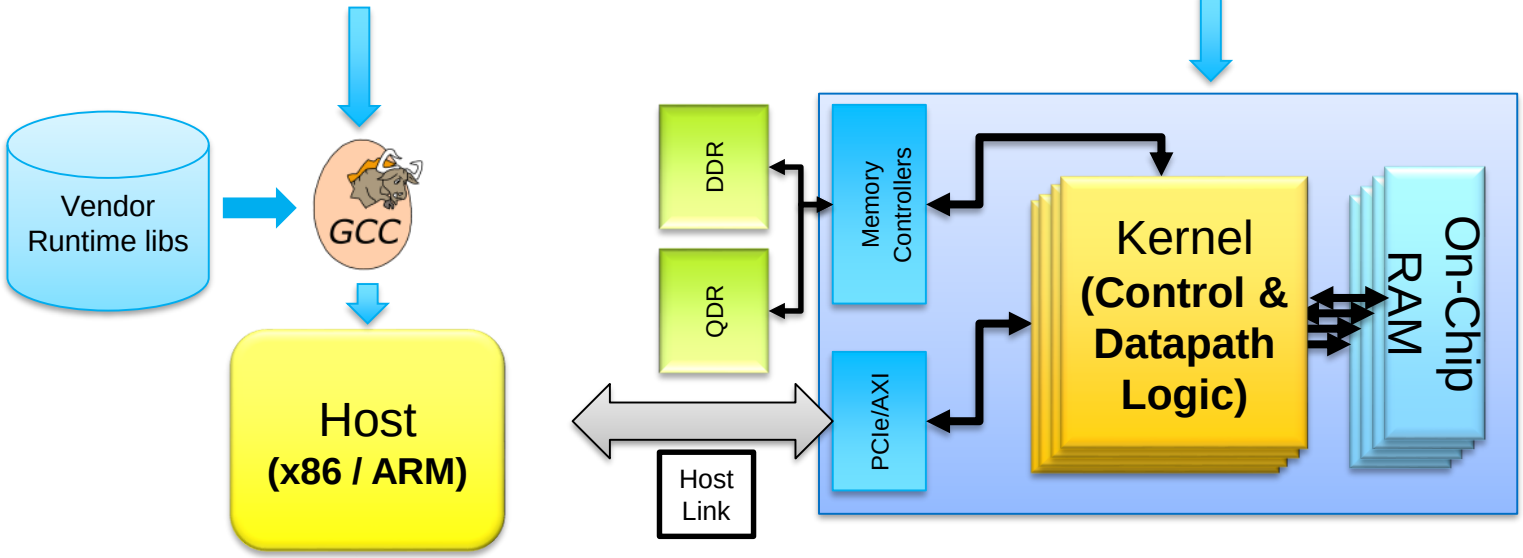
The Linux Kernel FPGA Manager Framework



Today's OpenCL Programming/Development Flow

```
main() {
   read_data( … );
   manipulate( … );
   clEnqueueWriteBuffer( … );
   clEnqueueNDRange(…,sum,…);
   clEnqueueReadBuffer( … );
   display_result( … );
}
```

```
kernel void
sum(global float *a,
    global float *b,
    global float *c)
{
  int gid = get_global_id(0);
  c[gid] = a[gid] + b[gid];
}
```



Reconfigurable Computing – Some Definitions

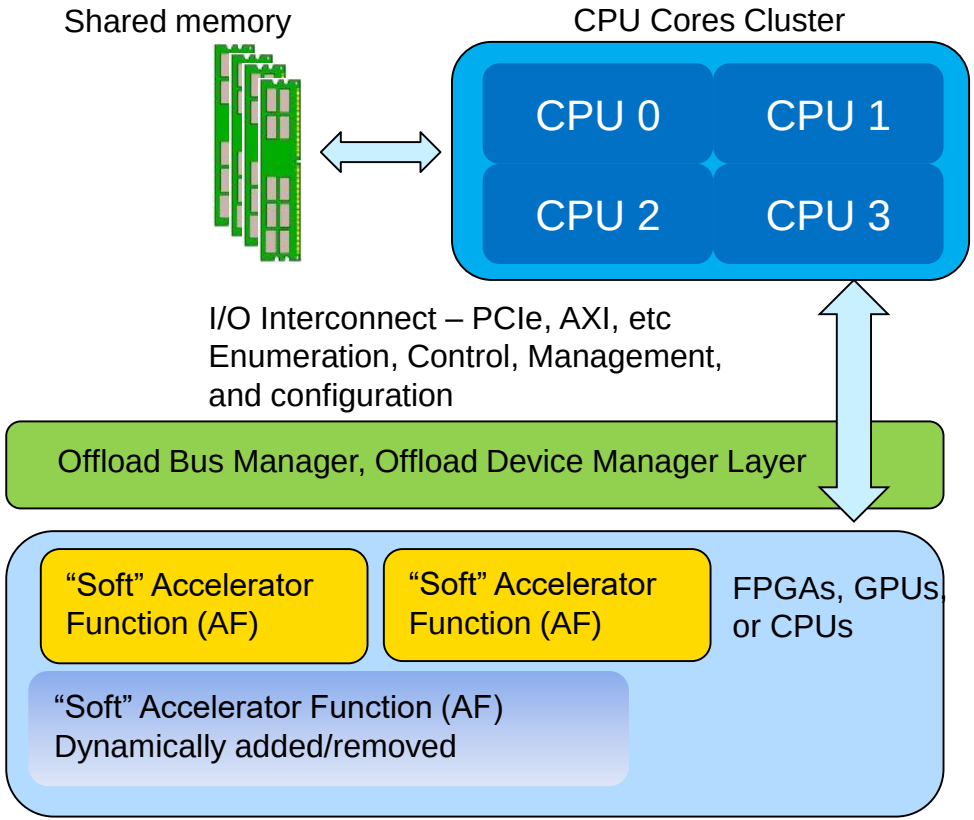
"CPU Cluster" – The group of CPUs running the Linux operating system.

"Accelerator Function (AF)" – A virtual device, created by programming a portion of the FPGA or one or more GPUs or CPUs.

"I/O Interconnect" – The technology used to attach the FPGAs, GPUs, or CPUs to the general purpose CPU cluster.

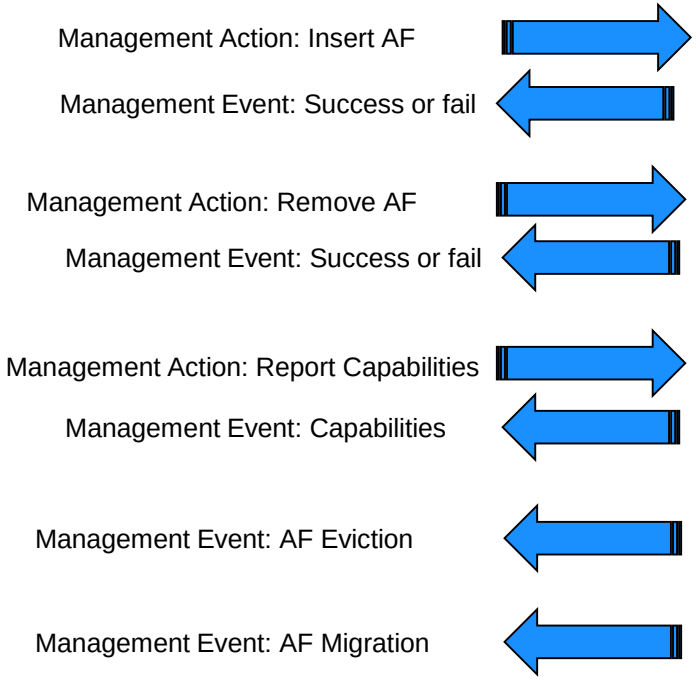
"Dynamic Insertion/Removal" – The process of adding/removing a "Soft Device" by programming the FPGA, GPUs, or CPUs.

"Resource Rebalancing" – The process of reallocating FPGA, GPU, or CPU resources by removing and reinserting "Soft Devices" to better use resources if needed.



Management Actions

- AFs and Programmable Device Resource Manager
  - Programmable devices have a finite mapped MMIO window and interrupts
  - AFs require certain amount of MMIO window and interrupts
- Insert/Remove/Suspend/Resume AF
  - AFs can be inserted, removed, suspended and resumed
- AF Eviction & Migration
  - AF may need to be forcefully evicted for higher priority AF, and could be “resumed” at some point in the future.
  - An AF may be migrated from one Programmable Device to another, which could be seen as an eviction from one device and insertion into another.
- Programmable Device Rebalance and Reconfiguration
  - As evictions and insertions occur, resources may need to be rebalanced.
- Miscellaneous Policy Management
  - AF priorities, affinity settings for processor affinity and physical I/Os.

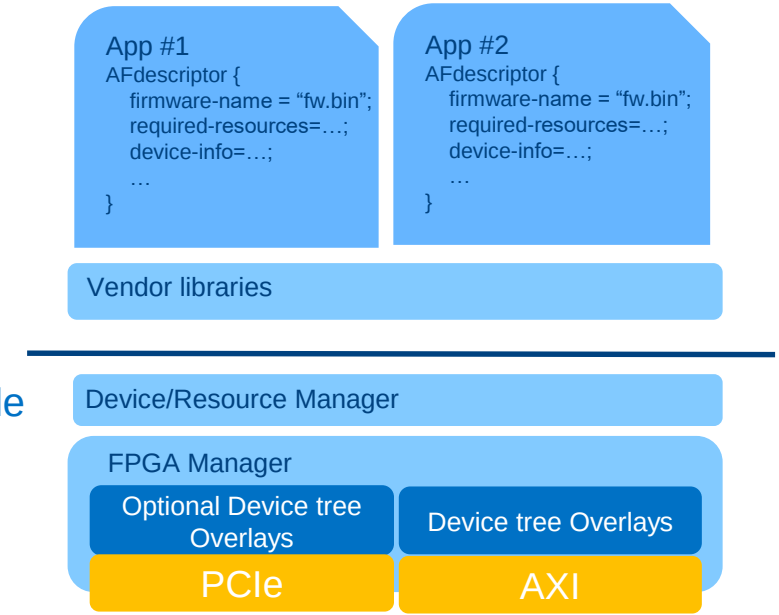


Device Discovery, Enumeration, Management

- PCIe and AXI are compared for Reconfigurable Computing Framework
- PCI Express
  - Used in Client/Server Systems for I/O Interconnects
  - Architecture supports management, configuration and discovery
- Advanced eXtensible Interface (AXI)
  - Used heavily in embedded systems using ARM processors and peripherals
  - No architectural support for device discovery – kernel uses device tree for static resource assignment and device discovery
- Framework should support these two types of I/O interconnects
  - Referred to as Discoverable and Non-discoverable.
- For this presentation, we focus specifically on
  - PCIe
  - AXI

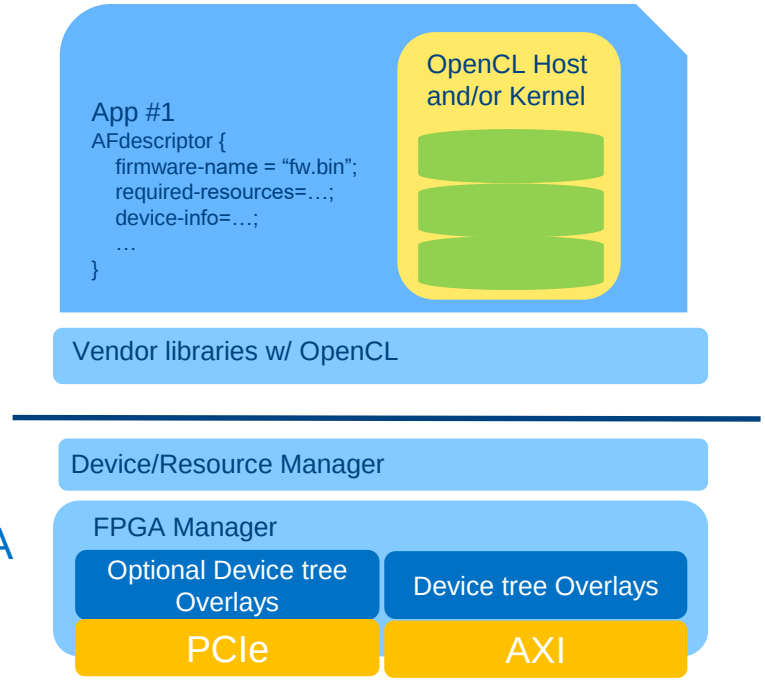
Device Example with PCIe, AXI using Device Tree Overlays

- Applications describe required resources in descriptor lists
  - Partial or complete configuration
  - I/O Resources, Interrupts (if any) required
  - Class of device (network, storage)
  - Transceivers, I/O pins required, etc.
  - Policies for configuration
- AF Descriptors are compiled to Device Tree Overlays
- Device Tree Overlays may be used for Non-discoverable interconnects (AXI for example). OF/ACPI.
- Device/Resource Manager finds matching FPGA with required attributes.
- FPGA Manager makes use of discoverable and non-discoverable interconnects.



Device Example using OpenCL

- Developer produces OpenCL host/kernel bundle using normal development process and tools.
  - OpenCL bundle is packaged with Application.
  - Vendor libraries recognize application as having an OpenCL bundle.
  - All described through descriptor lists
- AF Descriptors are compiled to Device Tree Overlays
- Device Tree Overlays may be used for Non-discoverable interconnects (AXI for example)
- Device/Resource Manager finds matching FPGA with required attributes.
- FPGA Manager makes use of discoverable and non-discoverable interconnects.



AF Descriptors

AF Descriptors contain information about the Acceleration Function required for the framework to instantiate the device.

- For each offload device needed {
  - Reference to FPGA program binary (the bitstream)
  - Expresses constraints (FPGA family, special resources, pins, etc)
  - Policy (Priority request, affinity, proximity CPU-FPGA interconnect)
  - List of devices (soft device, accelerator, memory map aperture, IRQ's, bindings)

  }

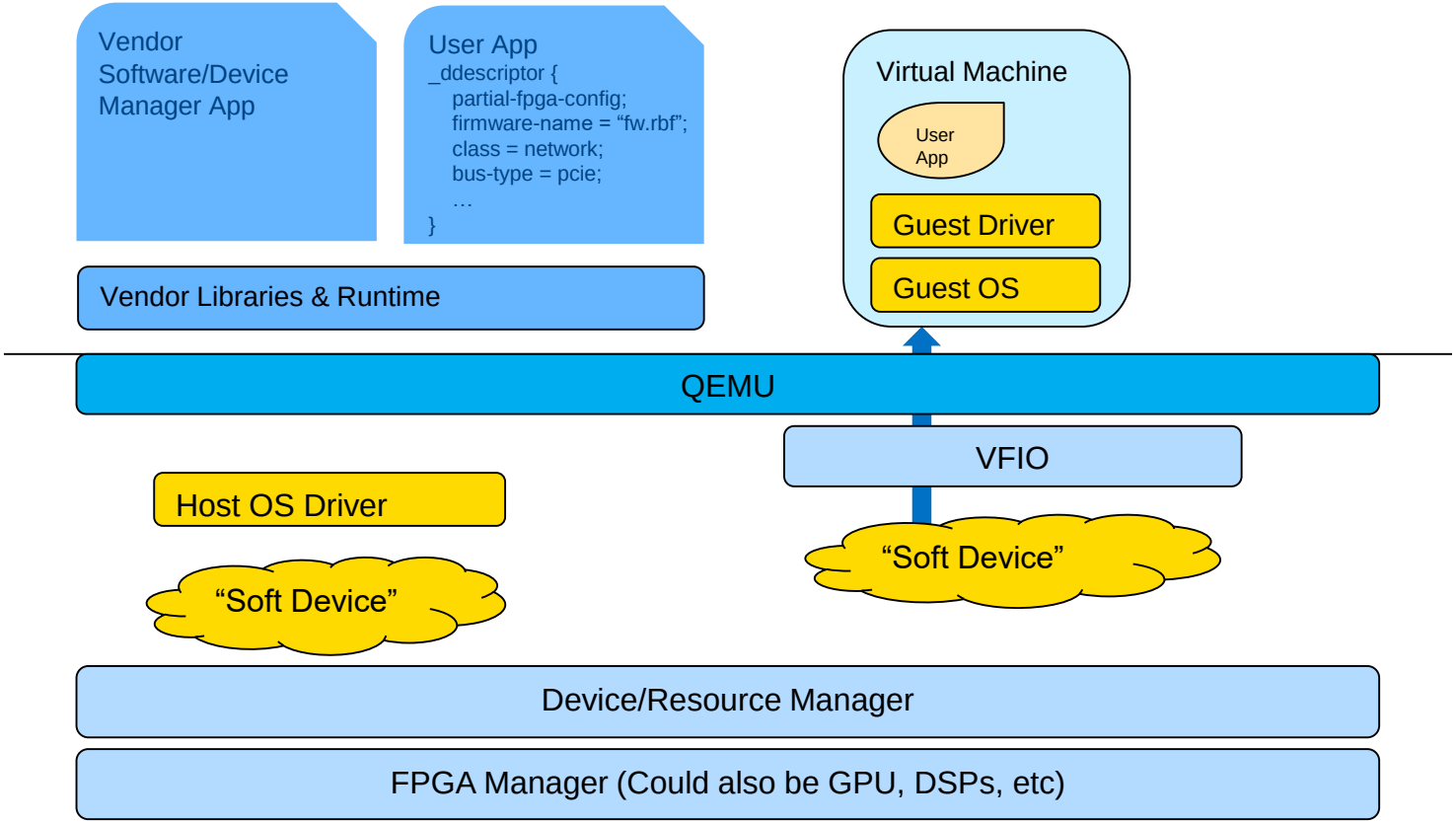
- Could contain nested blocks of descriptors

Resource Management Framework

Aka the libraries

- Vendor agnostic API, vendor specific plugins possible
- Receives descriptors from applications, translates in device tree in format appropriate for platform (OF, ACPI).
  - Requirements on contents is vendor specific
- Manage FPGA resources for conflicts. Etc
- Fed by policy/configuration
  - Permissions, usage time
  - Accounting is possible for implementing paid services

Hypervisor and Virtual Machine Support



Summary: Reconfigurable Computing Framework for Linux Requirements

- Must support Embedded Systems as well as Client/Server Systems
  - x86, ARM, any other CPUs that make use of offload elements
- Must support many types of interconnects
  - PCIe, AXI, etc
- Support different offload elements
  - GPUs, DSPs, FPGA, many integrated CPUs (MICs)
- Support existing technologies such as OpenCL – perhaps not directly
  - OpenCL and HLS used as embedded technology components within this framework
- Accelerator Function (AF) enumeration, resource management, dynamic AF insertion and removal, resource balancing
- Support exposure to AFs through a hypervisor and Virtual Machines

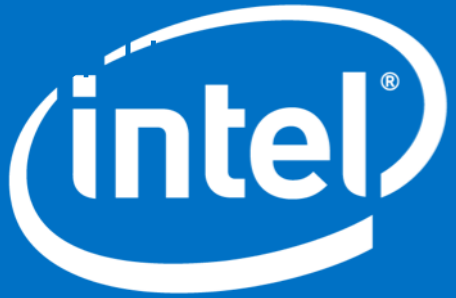
Acknowledgements

- Alan Tull, Findlay Shearer, Jun Nakashima, Susan Cohen, Mike Kinsner
- Linux Community
- Programmable Solutions Group (PSG) of Intel
- Linux Foundation

For more information…

- Reconfigurable Computing Group
  - http://wiki.linuxplumbersconf.org/2016:fpgas_and_programmable_logic_devices
- Email yves.vandervennet@intel.com and vince.bridgers@intel.com


ALTERA


intel